

Challenges in Applying Model-Based Systems Engineering: Human-Centered Design Perspective

So Young Kim, David Wagner, and Alejandro Jimenez

NASA Jet Propulsion Lab,
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109 USA
soyoung@jpl.nasa.gov

ABSTRACT

Systems engineering is, by its nature, human-centered. It is about tradeoffs and negotiations by systems engineers among different disciplines and systems to ensure the resulting system of systems can achieve the mission objectives successfully. Applying MBSE means enabling system engineers to use MBSE processes, methods, and tools to perform such work. This requires not only tools and infrastructure that support MBSE but also cultural change in how system engineers approach their work. Although the cultural change is listed as one of the challenges for MBSE infusion to address, its details and how to address them are not often discussed in the literature and the community. We used Human-Centered Design framework to articulate a core set of related challenges and the implications to tool and infrastructure developments. We will discuss the details of the HCD framework applied and discuss the core challenges using an electrical system engineering to harness engineering workflow as a case study.

Keywords

Human-centered design, Model-based system engineering, MBSE infusion

INTRODUCTION

Model-Based Systems Engineering (MBSE) is a systems engineering approach that was formulated to address the limitations of the traditional Document-Based Systems Engineering (DBSE) practice. While DBSE uses documents to convey the system design through the lifecycle of a project, MBSE uses system models that describe the relevant engineering elements and relationships between them (Chami and Bruel 2018) to convey the system design. As systems of interest become more complex in their mission objectives and composing elements, the limitations of the DBSE practices (such as ambiguity due to informal language, slow update cycles that cannot keep pace with the rapid evolution of a system, no efficient way to an automated validation, etc.) became evident. As a result, multiple US government organizations handling systems engineering projects (Young 2015, Wang, Izygon et al. 2016, Hale, Zimmerman et al. 2017) and other commercial companies such as Boeing and Thales, (Palmer, Crow et al. 2011, Asan,

Albrecht et al. 2014, Bonnet, Voirin et al. 2015) started to adopt MBSE as their systems engineering approach. MBSE has been touted to provide benefits such as better communication, better ways to manage complexity, better product quality, better ways to capture system knowledge, etc. (Friedenthal, Griego et al. 2007).

While adopting MBSE, a set of challenges has been identified and shared in the community from the perspectives of JPL (Estefan 2009, Fosse 2011, Cole 2013, Bayer 2018), other government agencies (Young 2015, Parrott 2016, Wang, Izygon et al. 2016, Hale, Zimmerman et al. 2017), other commercial sectors (Asan, Albrecht et al. 2014, Bonnet, Voirin et al. 2015), and systems engineering community (Chami, Aleksandraviciene et al. 2018, Chami and Bruel 2018). Some discuss based on their observations and experiences over the years (Carson 2015, Eisenmann 2015) at a philosophical level, and others discuss based on their day-to-day job experience (Jackson, Wilkerson et al. 2016, Wang, Izygon et al. 2016, Castet, Castillo et al. 2017). In these discussions, common challenges of applying MBSE are identified related to getting support from leadership, cultural change, specific infrastructure and tool development.

Model-Based System Engineering at JPL

At JPL, the systems engineering division recognized the need to adopt MBSE as a way to address ever-growing system complexity. The study conducted to identify a strategic vision and roadmap to apply MBSE (Estefan 2007) noted well-known challenges that are still discussed in other recent papers (Hale, Zimmerman et al. 2017, Vogelsang, Amorim et al. 2017, Chami, Aleksandraviciene et al. 2018, Chami and Bruel 2018). The vision of MBSE at JPL led to multiple MBSE-based projects including the Europa mission – now the Europa Clipper mission – (Bayer, Chung et al. 2012, Bayer, Chung et al. 2012, Bayer, Chung et al. 2013, Bayer 2018) and Mars 2020 (Fosse, Devereaux et al. 2015), as well as other research and application work such as ontology and modeling pattern for fault management (Castet, Rozek et al. 2015, Castet, Bareh et al. 2016, Castet, Bareh et al. 2018). Many lessons learned by the MBSE practitioners were shared with the internal and external

systems engineering community (Dubos, Schreiner et al. 2016, Jackson, Wilkerson et al. 2016, Castet, Castillo et al. 2017, McCoy, Nairouz et al. 2018).

MBSE inside Flight Projects at JPL

Europa Clipper, Mars 2020, Europa Lander among other projects applied MBSE in different levels of scale. In the case of Europa Clipper applied MBSE using the following team and infrastructure strategies. First, they developed a three-tier team structure: a small set of MBSE modeling experts (a.k.a., system modelers) within a larger set of domain expert systems engineers, within a larger set of all project personnel (Bayer, Chung et al. 2012). SysML-based tooling environment mixed with COTS and in-house tools was chosen. The system modeler who is fluent in SysML collaborates with the domain expert and models the system of interest (i.e., describe system data in SysML) using a COTS tool. The system modeler generates system engineering products (e.g., mass roll-up table), and the domain experts can view those system engineering products using an in-house web-based tool. In the case of Mars 2020, they implemented a two-tier team structure which involved the training the systems engineers themselves to develop their own system models directly with support as needed from modeling experts. This was enabled by developing and applying internally developed MBSE tools and processes.

A Point of View from MBSE Practitioners' Perspective (Castet, Castillo et al. 2017)

A group of system modelers and systems engineers who were part of both the Europa Clipper and Mars 2020 flight projects published an internal presentation on the challenges that they faced along with suggestions for how to address them. General observations were in two categories: 1) introduction of MBSE approach, extensive tool development, and implementation of new processes all at the same time was quite taxing on day-to-day activities of system modelers and domain experts and 2) immature tooling and tool development without clear understanding of users' work and workflow impacted the project work negatively. Another noteworthy perspective was that the system modelers themselves believed that having dedicated system modelers to pair with domain experts was a good starting point but not a viable option for future projects. They believed it cannot efficiently scale to a whole project without increased labor cost and creation of bottlenecks to enter system data into the model. Also, it creates unnecessary validation work because the owner (author) of the information is not the one entering it to the system model.

Human-Centered Design

Human-Centered Design (HCD) is a design framework that considers technologies, people's activities, and organizations (TPO) when designing a complex socio-technical system using different kinds of scientific methods (Boy 2017). For example, when applied to a product design, the HCD framework considers 1) is it technically feasible, 2) is it what people needs, and 3) is it viable for the business. The HCD framework provides a systematic and comprehensive way to involve every stakeholder (e.g., users,

managers, builders, maintainers, etc.) of a system of interest into the design. We applied this framework to identify underlying challenges of infusing MBSE at JPL.

CASE STUDY: Electrical Systems Engineering to Harness Engineering Workflow

Participants

Total of 14 subject matter experts (14 male and 1 female) were recruited to participate in the research phase. They were part of the research to 1) understand the current MBSE implementation at JPL and 2) to understand the workflow of the systems engineering practitioner users. Of those, 12 participants currently work on active flight projects. 10 participants have expertise in electrical systems engineering, one in instrument system, and another one in fault protection. We used Competencies Proficiency Scale (NIH 2019) to understand their level of experience and expertise in 1) subject matter domain and 2) MBSE. The Competencies Proficiency Scale (CPS) is a 5-level Likert scale mapped to Fundamental Awareness (basic knowledge), Novice (limited experience), Intermediate (practical application), Advanced (applied theory), and Expert (recognized authority) in which 1 being Fundamental Awareness and 5 being Expert. The average CPS for subject matter domain was 3.39 with standard deviation 0.83, the one for MBSE was 2.68 with standard deviation 1.45. This indicates that the participant distribution was fairly even in their level of expertise around Intermediate, but the distribution in their level of experience in MBSE was wide, indicating some participants had no experience in MBSE while others had quite extensive experience in MBSE.

We used a set of HCD research methods such as one-on-one semi-structured interviews, task walkthrough, focus group, product walkthrough, etc. to engage these stakeholders.

Electrical Systems Engineering

Electrical Systems Engineers (ESEs) in a flight project at JPL ensure that all electrical system components of subsystems and instruments can be integrated together. They especially work closely with HEs so that the resulting harness design meets the electrical functional needs. **Error! Reference source not found.** illustrates communications and coordination among different groups in a flight project around ESEs.

Error! Reference source not found. illustrates the systems engineers' workflow in a traditional flight project process. ESEs create an Excel spreadsheet to describe their system design at an electrical functional level based on project design documents from subsystems. Among other things, ESEs generate margin reports on electrical resource required and available (e.g., number of power switches or temperature sensor interfaces available), using Excel spreadsheet. When the electrical system design reaches a certain level, ESEs generate electrical circuit data sheets (ECDSs) that describe detailed circuitry and connections as an input to HEs. These are in a PDF format, and manual "red-lining" iterations between ESEs and HEs have to occur as a means to communicate their designs, requirements, and

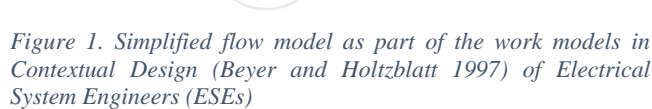
The diagram illustrates the role of the Electrical Flight System Engineer (EFSE) as a central hub. The EFSE is represented by a central icon of a person at a computer, labeled "Electrical Flight System Engineer (EFSE)".

Surrounding the EFSE are several teams and their interactions:

- Other Flight Systems Engineers:** Located at the top, connected to the EFSE by a solid black arrow pointing upwards.
- Flight Systems SE:** Located below the Other Flight Systems Engineers, connected to the EFSE by a solid black arrow pointing downwards.
- Flight System Block Diagram:** Located below the Flight Systems SE, connected to the EFSE by a solid black arrow pointing downwards.
- Project System Engineering Team (PSET):** Located to the left, connected to the EFSE by a dashed black arrow pointing leftwards.
- ATLO:** Located at the bottom left, connected to the EFSE by a dashed black arrow pointing leftwards.
- 6X Electromagnetic Compatibility:** Located at the bottom, connected to the EFSE by a dashed black arrow pointing downwards.
- Cable & Harness Subsystem Cog Engineer:** Located to the right, connected to the EFSE by a solid orange arrow pointing rightwards.
- Other Subsystem Cog Engineer:** Located at the bottom right, connected to the EFSE by a solid orange arrow pointing rightwards.

Additional labels and connections include:

- Reference Designator Function List Circuit Data Sheet:** A blue dashed arrow pointing from the EFSE to the Cable & Harness Subsystem Cog Engineer.
- Harness Block Diagram:** A solid orange arrow pointing from the Cable & Harness Subsystem Cog Engineer to the EFSE.
- Interface Circuit Data:** A solid orange arrow pointing from the Other Subsystem Cog Engineer to the EFSE.



The diagram illustrates the design and development of a system, divided into two main phases: Design Electrical Aspects of System and Design Physical Aspects of System.

Design Electrical Aspects of System

Other Subsystem SEs

ESE

1. Assemble SD, Configure LUT, Configure I/O

2.1. Validate that all the components are assigned

3. Analyze net list based on the existing physical design

4. Review the Proposed CS SD, Configure I/O

5. Identify resources at function of the FS

6. Create/Update Resource Investigator

7. Create/Update Function List

8. Create/Update Electrical System BU

9. Validate sources when changes are introduced

Design Physical Aspects of System

10. Add and route

11. Create/Update Electrical Circuit Data Sheet

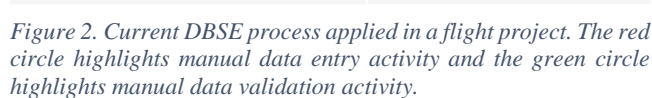
12. Synthesize and validate net and BOM

13. Create/Update Harness Block Diagram

14. Create/Update Harness Block Diagram

15. Create/Update Harness Fabrication Diagram

16. Create/Update Harness Fabrication Diagram



CHALLENGES IN APPLYING MBSE AT JPL FROM HCD PERSPECTIVE

How Systems Engineers Conceptualize and Specify their Systems

The systems engineers often conceptualize their systems within the limitations of their legacy tools, processes, and deliverables that are developed and matured in a typical DBSE approach. The legacy tools often require them to conceptualize the system design in one way but to describe in a different way as illustrated later in the section. Consequently, this discrepancy results in ambiguous specifications of the system design because significant amount of contextual data is not described formally, but they are implied through tribal knowledge such as conventions, unspoken rules by the community, etc.

In our case study, in the beginning of the modeling process for the electrical systems, ESEs typically start with an Excel spreadsheet template. This template usually evolves as the

1. Different levels of abstractions are expressed in the same view, allowing electrical functional design, net design, wire design all to be described in one view. This becomes problematic because the topology of relations may be different at different levels of abstraction.
2. Different kinds of data are expressed in rows of the same table failing to normalize information.
3. Instead of providing requirements to HEs, actual designs of harness are described. This frequently over-constrains the solution, leading to time-consuming, error-prone, unnecessary re-work between ESEs and HEs. In addition, since this design detail is not authoritative in the specification, it becomes inconsistent with the final design, making design validation difficult.

4. Same data is modeled (i.e., entered into the tool) multiple times. For example, defining an interconnection between a heater and a power switch card implies the existence of a power switch. However, the characteristics of that power switch is captured separately in the spreadsheet. This may create an inconsistency in the system description.

These issues are handled and managed by dedicated systems engineers to ensure the integrity of the system design. They go through overwhelming amount of manual validation and re-work to make their electrical system meets its requirements. This situation, however, makes the systems engineers to waste their engineering time on manual data entry, manual data checking, and unnecessary coordination. Also, more importantly, it ingrains the systems engineers to describe their system within this structure. This mindset of systems engineers is one of the most important challenges to address.

Perception that a Tool Does MBSE

Even though there are multiple papers discussing MBSE as more than just a tool (Estefan 2009, Bonnet, Voirin et al. 2015, Young 2015, Wang, Izygon et al. 2016, Chami, Aleksandraviciene et al. 2018), when the discussions about how to infuse and apply MBSE occur, the conversation tends to become which tools should be developed and deployed. This is because the pervasive implicit thinking is that MBSE is enabled by providing a tool that “does MBSE.”

However, a tool does not do MBSE. Tools enhance methods and processes. Systems engineers do MBSE by applying formal methods and model-based processes enabled by dedicated tools. Establishing model-based formal methods is difficult to address because it depends on the systems

engineering processes and conventions of the systems engineering community in a company. As emphasized in the challenge above, most systems engineers who do hands-on work are not familiar with the MBSE approach, and they tend to describe their system design (i.e., the method) to fit within the tools and processes that are mandated/available to them. Without properly-structured model data, applying MBSE yields limited benefits. Even worse, it can generate inconsistent systems engineering products that could threaten the integrity of the system design.

Migration to MBSE should not only emphasize the tool selection and development but also, more importantly, emphasize establishing model-based systems engineering methods that are tailored to each discipline.

Usability Issues Masking the Underlying Challenges

When currently-available MBSE-enabling tools were introduced to and used by the systems engineering community at JPL, one of the most talked-about issues was usability. Too many aspects and attributes had to be specified to describe a simple system characteristic (a.k.a., too many clicks). The usability issue is indeed a valid concern and an important one. From the HCD perspective, we all recognize how crucial this issue is especially for new technology adoption. If the tool requires significantly more effort to describe the system compared to an existing tool (e.g., Excel), it is easy to predict that many systems engineers will not adopt the tools and by corollary, MBSE methods and processes.

However, this issue masked another important challenge. The semantics of SysML do not lend themselves well to directly describe specific discipline attributes that systems engineering at JPL want to capture. However, for these attributes to be captured in the model, sufficient precision and formalism needed to be infused into their systems engineering processes, which is currently lacking and impedes the modeling work of domain experts. As such, the usability issues of an MBSE tool should not be solely blamed for difficulties in wider MBSE adoption in the systems engineering community.

Therefore, our complete answer to this problem should be neither the most intuitive generic MBSE tool nor an Excel or Excel-like tool alone. Without addressing the challenge to formalize the discipline methods together with the usability issues, we cannot truly achieve the benefits of MBSE.

Custom Tools are built or COTS Tools are Selected without Deep User Understanding.

When tools are built or selected for MBSE infusion, the design or adoption decisions are made based on what capabilities that the tools should or can provide based on a one-dimensional view of user needs. Identifying user needs and other user engagement efforts exist, however, they lack in covering the full lifecycle users' work or full scope of scale of operations that users need to do.

Satisfying a set of needs based on one point in a project lifecycle is unlikely to support the whole project throughout its lifecycle. For example, having construct-by-correction to

enforce the correctness and completeness of the system model is important, but the tools should also support the early design stages in which most aspects of the system design are not defined and detailed, yet.

Selecting a tool because it is generic and being used across industries without fully exercising it for the intended operations that users need to do can lead to serious issues preventing adoption. For example, testing a tool using a "test model" which is drastically simplified with a limited number of test users often fails to capture the full scope of operational requirements, and deployment can result in significant performance issues.

ADDRESSING CHALLENGES

Successful MBSE infusion requires MBSE tools (Technology in TPO), MBSE-minded system engineers (People in TPO), and MBSE processes and methods (Organization in TPO). Ergo, MBSE Capability Development should balance its focus among 1) Defining MB Processes & Methods 2) Training and growing people with MB mindset and 3) Developing tools that supports the processes and methods (useful) and people (usable).

Computer Aided Engineering for Space Systems Architectures (CAESAR) for MBSE at JPL

CAESAR is an integrated SE tool suite with methodologies that support the transformation of current SE practices to model-centric engineering practices.

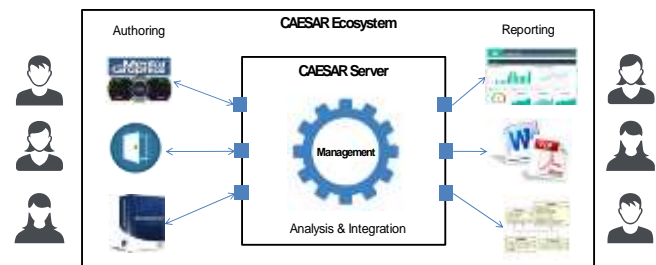


Figure 3 CAESAR provide infrastructure and applications that supports MBSE formal methods that are co-developed with the owners of the discipline processes and methods.

Development Approach

The strategy of CAESAR development is tightly mapped to the TPO framework: 1) focus on a specific discipline at a time (*T*), 2) focus on use cases for current flight projects (*O*), 3) Empower and support line management (owners of the discipline processes and methods) to define discipline-specific MBSE methods. (*O*), 4) Co-design the tools with the user community and use that as an MBSE infusion opportunity (*TP*), and 5) Enable systems engineers to describe their systems, using their terminologies (no "system modeler," *TPO*).

Development Process

Error! Reference source not found. illustrates CAESAR's approach comprising of two main efforts. The first is to translate traditional systems engineering methods and processes into model-based ones, chartered to Product Systems Engineering team, and the second is providing tool ecosystem that supports the model-based methods and

processes, chartered to the Product Design and Development team.

The Product Systems Engineering team focuses on working with domain experts to understand and document current workflows and codify existing system engineering methods and processes. Note that the specifications for developing model-based methods needs to be detailed and precise. For example, the team needed to understand not only what a single data point in their current spreadsheet refers to but also where this data point comes from, how it is authored, how it is related to the rest of the system model. Without this level of understanding, we could not define a proper way to structure the patterns of abstraction and relationship into a model-based method.

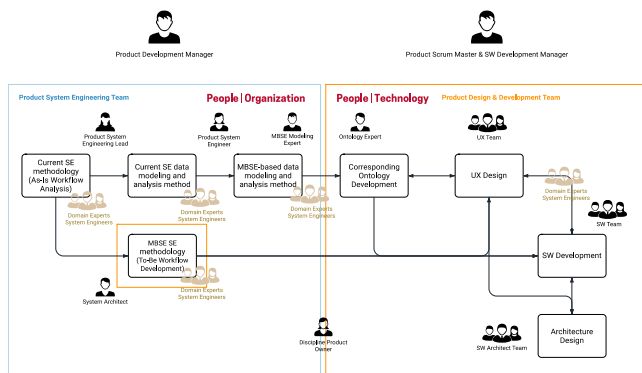


Figure 4. CAESAR's MBSE infusion process based on HCD thinking

The team also works on ontology development to contextualize a modeling language into the domain of interest. CAESAR's Modeling/Ontology experts work with domain experts to map and conceptualize the system model formally. Note that this team structure differs from the tiered-team structure used in Europa Clipper or Mars 2020. The modeling experts' tasks are not to enter data or generate engineering products for the domain experts. In CAESAR's approach, the modeling experts collaborate closely with the domain experts to establish a modeling pattern consistent with the domain-specific ontology. The domain experts are thus empowered to author the model data themselves. This process has been effective in influencing traditional system engineers' attitude towards how to describe their model data unambiguously and formally within the modeling language structure.

The Product Design and Development team focuses on designing an overall product architecture and products including platform and applications based on HCD and agile approaches. The details of the platform and the applications are out of the scope of this paper. The interested readers can find an overview in OpenCAESAR information page (CAESAR 2019).

The CAESAR team successfully launched its operational application for the ESE team in Europa Clipper based on this process. The CAESAR's Electrical Flight System Engineering application provides the modeling workbench

in which ESEs describe the system design in their domain-specific language in a correct-by-construction format with a flexibility to "sketch their design." Also, the application provides a web-based application where ESEs can initiate ontological validation and integration of their models. It also generates the engineering products based on the system model data, shared across the project.

DISCUSSION

The efforts from Product Systems Engineering team enabled not only uncovering the underlying challenges but also influencing behavioral changes in how domain experts approach system design. Anecdotally speaking, in the beginning of the engagement, the participating ESEs would describe their system using data specified in a column of a spreadsheet. The interaction between the domain experts and the modeling experts enabled the changes in the domain experts' behavior in that they started to specify their design as a property of an electrical function instead of a column in a spreadsheet. The team observed that the domain experts describe their design challenges in a model-based manner.

The next question is how to scale this change up to a companywide MBSE infusion, which falls under organization change management. Because organization change management is a discipline on its own, it will be not discussed in details here. A recent article published by McKinsey & Company (Hamel and Zanini 2014) summarized the limitations of both top-down and bottom-up approaches: the former with 70% failure rate on top-down approach change initiative and the latter not resulting in systematic changes across an entire organization. Infusing MBSE companywide shares same challenges. An MBSE infusion effort should apply both approaches to ensure MBSE processes and methods are defined and practiced throughout the organization and to grow change agents at the practitioner level who can spread directly-experienced benefits to other practitioners.

CONCLUSION

Successful infusion of MBSE requires more than selecting and developing tools and infrastructure. It requires: 1) deep understanding of existing system engineering practices, gaps, and limitations, 2) investment on developing domain specific ontologies that precisely capture the particular discipline abstractions, 3) defining MBSE methods and processes so that domain experts can describe their systems correctly and meaningfully using the ontologies, and 4) corresponding tools and infrastructure.

Systems engineering is, by its nature, human-centered. It is about tradeoffs and negotiations among different disciplines and systems to ensure the resulting system of systems can achieve the mission objectives successfully. Applying MBSE aims to do systems engineering more effectively, with less reinvention of ideas, methodologies, or tools. At the same time, it aims to reduce both process and product risks by ensuring a more precise, complete, and centralized specification of the system design. Therefore, MBSE infusion is not just a technology infusion: it is a cultural and

technological paradigm shift, requiring efforts that need to be deeply human-centered.

ACKNOWLEDGMENTS

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Copyright 2019 California Institute of Technology. U.S. Government sponsorship acknowledged.

The authors would like to thank Dr. Jean-Francois Castet for his reviews and comments to shape this paper more clearly and other reviewers. Lastly, we thank our user group for their support and engagement working with the CAESAR team.

REFERENCE

- Asan, E., et al. (2014). Handling Complexity in System of Systems Projects—Lessons Learned from MBSE Efforts in Border Security Projects. *Complex Systems Design & Management*, Springer: 281-299.
- Bayer, T. (2018). Is MBSE helping? Measuring value on Europa Clipper. 2018 IEEE Aerospace Conference, IEEE.
- Bayer, T., et al. (2012). "Early formulation model-centric engineering on NASA's Europa mission concept study."
- Bayer, T. J., et al. (2012). Model based systems engineering on the Europa mission concept study. Aerospace Conference, 2012 IEEE, IEEE.
- Bayer, T. J., et al. (2013). Update on the model based systems engineering on the Europa mission concept study. INCOSE International Symposium, Wiley Online Library.
- Beyer, H. and K. Holtzblatt (1997). Contextual design: defining customer-centered systems, Elsevier.
- Bonnet, S., et al. (2015). Implementing the MBSE Cultural Change: Organization, Coaching and Lessons Learned. INCOSE International Symposium, Wiley Online Library.
- Boy, G. A. (2017). "Human-centered design of complex systems: An experience-based approach." *Design Science* 3.
- CAESAR (2019). "Open CAESAR Enabling Model Centric Engineering." from <https://open-caesar.github.io/>.
- Carson, R. S. (2015). "MBSE Implementation across Diverse Domains." *Insight* 18(2): 22-23.
- Castet, J.-F., et al. (2016). Fault Management Ontology and Modeling Patterns. AIAA SPACE 2016: 5544.
- Castet, J.-F., et al. (2018). Failure analysis and products in a model-based environment. 2018 IEEE Aerospace Conference, IEEE.
- Castet, J.-F., et al. (2015). Ontology and modeling patterns for state-based behavior representation. AIAA Infotech@Aerospace: 1115.
- Castet, J. F., et al. (2017). A point of view from MBSE practitioners, NASA JPL.
- Chami, M., et al. (2018). Towards solving MBSE adoption challenges: the D3 MBSE adoption toolbox. INCOSE International Symposium, Wiley Online Library.
- Chami, M. and J.-M. Bruel (2018). "A Survey on MBSE Adoption Challenges."
- Cole, B. (2013). "JPL Community View on Challenges and Rewards of MBSE."
- Dubos, G., et al. (2016). Architecture Modeling on the Europa Project. AIAA SPACE 2016: 5310.
- Eisenmann, H. (2015). "MBSE Has a Good Start; Requires More Work for Sufficient Support of Systems Engineering Activities through Models." *Insight* 18(2): 14-18.
- Estefan, J. (2007). A Strategy and Roadmap for Advancing the State-of-the-Practice of Model-Based Engineering at JPL: A Systems and Software Division Perspective. NASA JPL.
- Estefan, J. (2009). "MBSE methodology survey." *Insight* 12(4): 16-18.
- Fosse, E. (2011). "Model-based systems engineering (MBSE) 101."
- Fosse, E., et al. (2015). Inheriting Curiosity: leveraging MBSE to build Mars2020. AIAA SPACE 2015 Conference and Exposition.
- Friedenthal, S., et al. (2007). INCOSE model based systems engineering (MBSE) initiative. INCOSE 2007 Symposium.
- Hale, J., et al. (2017). "Digital Model-Based Engineering: Expectations, Prerequisites, and Challenges of Infusion."
- Hamel, G. and M. Zanini (2014). "Build a change platform, not a change program." from www.mckinsey.com.
- Jackson, M., et al. (2016). Exposing hidden parts of the SE process: MBSE patterns and tools for tracking and traceability. Aerospace Conference, 2016 IEEE, IEEE.
- McCoy, K., et al. (2018). Assessing the science robustness of the Europa clipper mission: Science sensitivity model. 2018 IEEE Aerospace Conference, IEEE.
- NIH (2019, January 12, 2009). "Competencies Proficiency Scale." 2019, from <https://hr.nih.gov/working-nih/competencies/competencies-proficiency-scale>.
- Palmer, J. R., et al. (2011). Model-Based Systems Engineering without SysML. Boeing presentation to National Defense Industrial Association Systems Engineering Conference, San Diego.
- Parrott, E. (2016). "The Value of Successful MBSE Adoption."
- Vogelsang, A., et al. (2017). "Should I stay or should I go? On forces that drive and prevent MBSE adoption in the embedded systems industry." arXiv preprint arXiv:1709.00266.
- Wang, L., et al. (2016). Effort to Accelerate MBSE Adoption and Usage at JSC. AIAA SPACE 2016: 5542.
- Young, K. G. (2015). Defense Space Application of MBSE-Closing the Culture Chasms. AIAA SPACE 2015 Conference and Exposition.